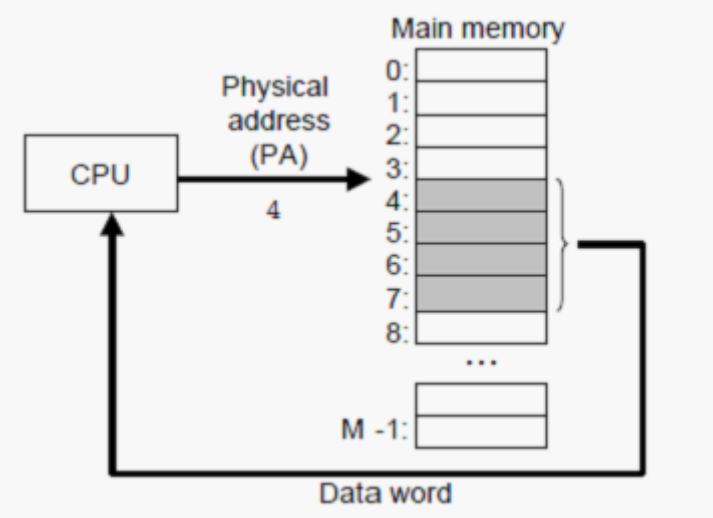
内存计算之内存管理

笔记本:	内存计算		
创建时间:	2018/10/22 9:13	**更新时间:**	2018/10/29 12:01
作者:	simba_wei		

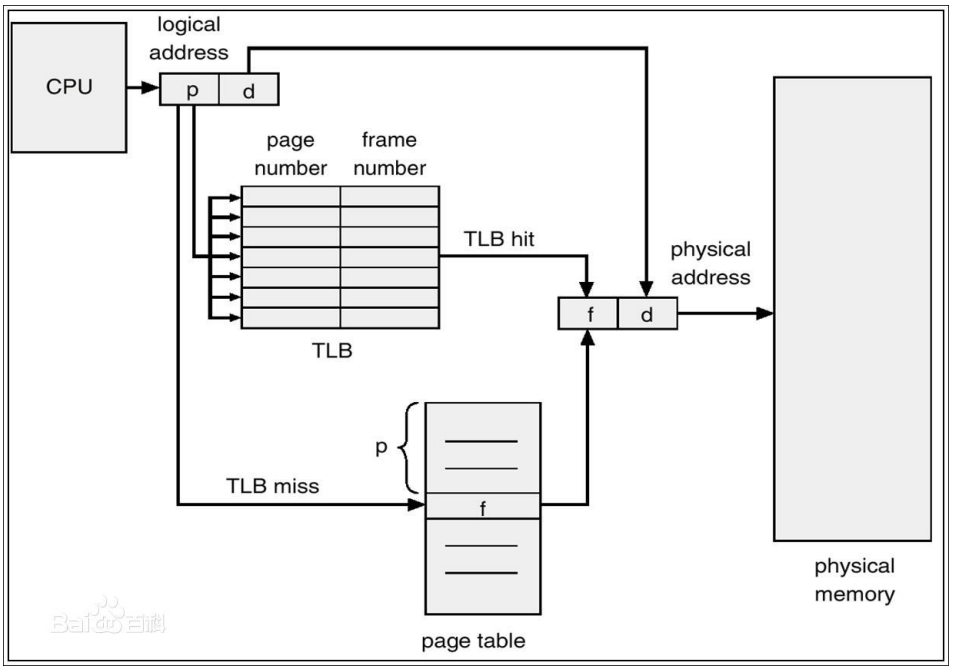
基于单节点的内存管理

单节点机器中的内存可以是一片集中式的共享内存，也可以是按照CPU来分配的非一致性共享内存(non-uniform memory access，简称NUMA)。
这些内存可以通过一种统一的机制进行管理，我们称之为“虚拟内存”。它的作用是对应用上层将物理内存封装出一片连续的内存空间，使得上层可以去调用一片连续的内存空间（空间上可能是碎片化的内存空间）。这种较好的提升内存的利用率的方法被称为“虚拟寻址”（virtual addressing）。然而，这种方式不同于我们在操作系统课程上学到的“虚拟内存”（virtual memory）这一概念。虚拟内存的提出是为了应对计算机应用的体量日益庞大，对内存的需求也日益庞大。而传统机器中的内存增长速度往往不能符合当前应用的需求，这就需要一套机制实现磁盘到主存空间的映射。下面我们分别讨论一下这两个概念

virtual addressing：真实的物理地址意味着程序实际上是知道了机器上RAM的布局。例如，当我们GDB调试程序是，我们发现访问的某个变量的地址为 0x8746b3时，即表示该变量在RAM中的真实的物理地址。使用虚拟寻址，所有应用程序内存访问都会转到页表，然后该表会从虚拟地址映射到物理地址。所以每个应用程序都有自己的“私有”地址空间，并且任何程序都不能读取或写入另一个程序的内存。这就是所谓的*分割*。虚拟寻址有许多好处。它通过糟糕的指针操作等保护程序不会互相碰撞。因为每个程序都有自己独特的***虚拟内存集***，所以程序不能读取其他的数据。 这是安全性。虚拟内存还支持***分页****功能*，程序的物理内存可能在不使用时存储在磁盘上（或现在更慢的闪存。这就是我们所说的virtual memory），***然后在应用程序尝试访问该页面时被调回***。这一策略就是将RAM看做一个介于CPU和磁盘之间的一个层地址空间的高速缓存。在进一步了解虚拟寻址之前，我们先了解一下***物理地址(physical address)***和***虚拟地址(virtual address)***。物理地址：计算机系统的主存被组织成一个由M个连续的字节大小的单元组成的数组，每个字节都有一个唯一的物理地址。



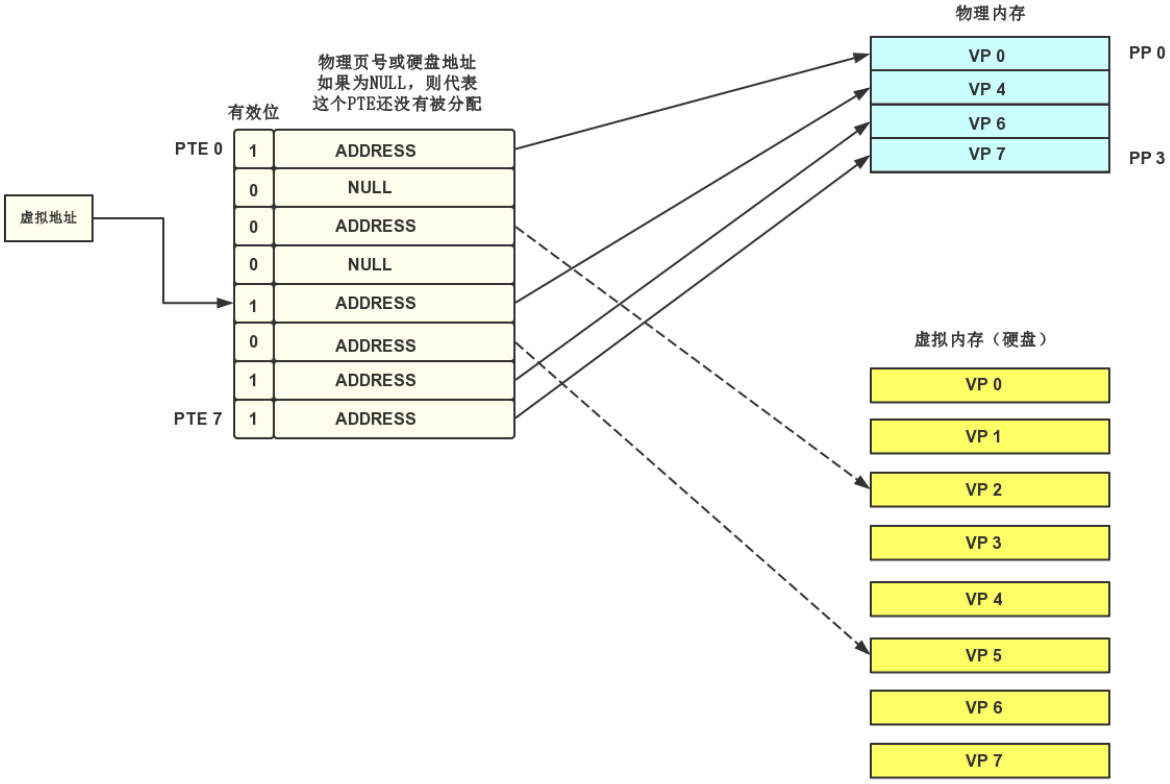
上图就很好的说明了CPU直接访问RAM中数据的方式，其中RAM数组中每一位都是一个8位字节（大都数）。而图中实际上是省略了一层“地址总线（address bus）”机制，其中内存总线的“宽度”就决定了CPU的寻址范围，及能够支持的内存大小。

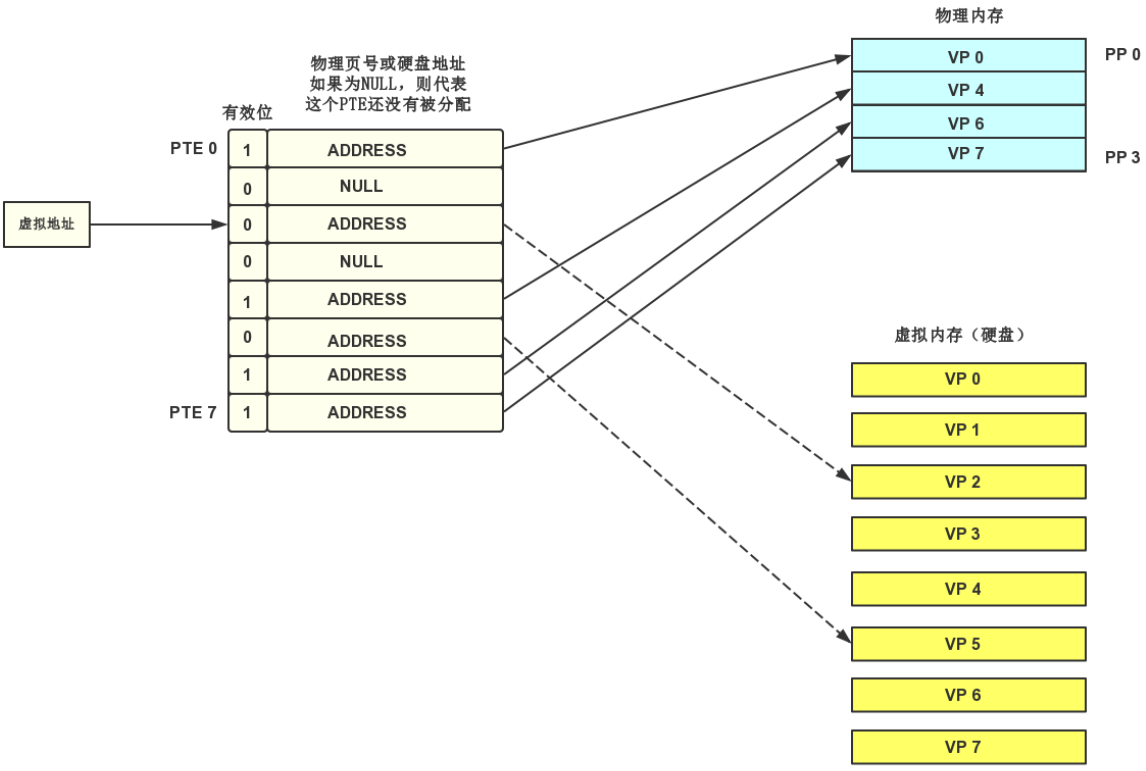


上图就是虚拟寻址方式的描述。对于一个程序，CPU直接拿到的是虚拟地址。CPU需要将这个逻辑地址输入到MMU单元中从而计算出在RAM中的真实地址。其中，MMU（memory management unit）中的页表（page table）记录了虚拟页号和RAM的帧号对应的关系。假设一片RAM中每一帧的大小为M位，而记录页表的数组的一条entry的大小为N位，那么每个entry里面的高N-M位用来表示对应的具体的帧号，低M为表示偏移量。CPU和MMU之间还存在着一层缓存buffer（TLB），CPU拿着逻辑地址先去访问TLB中的映射表，根据逻辑地址直接拿到对应的页号和帧号。

virtual memory

在缓存原理中，换入/换出的数据以块为最小单位。在内存管理时，页是地址空间的最小单位。虚拟地址空间划分为多个固定大小的虚拟页(VP),物理地址空间(DRAM内存)划分为多个固定大小的物理页(PP), 虚拟页和物理页的大小是一样的，通常为4KB。虚拟页和物理页存在着以下关系：虚拟页和磁盘文件映射，然后缓存到物理页。根据是否映射，是否缓存，可以将虚拟页的状态分为以下三种： 1)未映射的页即虚拟页没有映射到磁盘文件；2)未缓存的页虚拟页映射到了磁盘文件，但是没有缓存到物理页，也就是内存上；3)缓存的页虚拟页映射到了磁盘文件，并且缓存到物理页。如下图所示：





我们要清楚，当一个线程调用系统的动态内存分配器去给当前的任务动态的分配内存时，这片分配的内存时对应着虚拟内存区域，也就是我们所熟悉的堆（heap）

基于分布式系统的内存管理